

# **Introduction to SAS Procedures**

**ICER Biostatistics Unit  
February 2001**

Presented by: Tara Dudley, Mstat  
Amy Jeffreys, Mstat

Website: [hsrd.durham.med.va.gov/Biostat/](http://hsrd.durham.med.va.gov/Biostat/)

# Introduction to SAS Procedures

Version 6.12

- SAS data set information

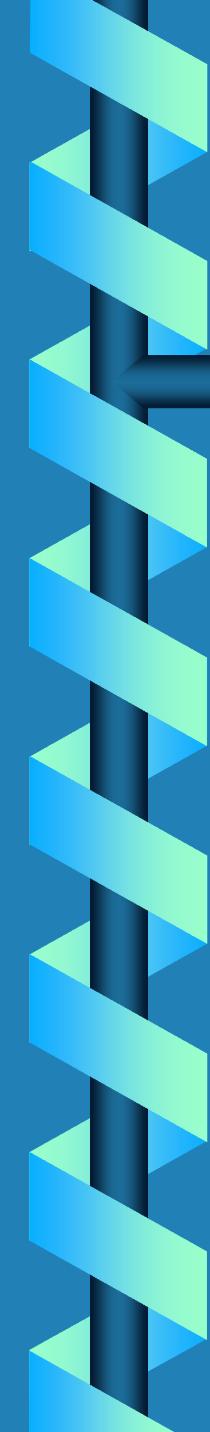
- PROC CONTENTS
  - PROC PRINT

- Descriptive statistics

- PROC MEANS / PROC SUMMARY
  - PROC UNIVARIATE
  - PROC FREQ

- Simple plots

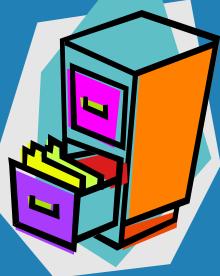
- PROC PLOT



# What does my SAS Data Set Contain?

---

- ➊ How many observations?
- ➋ How many variables?
- ➌ What kind of variables?

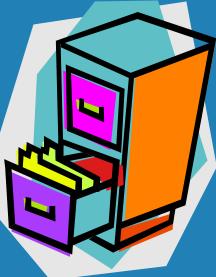


# PROC CONTENTS

- Provides information about the contents of a SAS data set

- Syntax:

```
PROC CONTENTS DATA=data set name;  
RUN;
```



# PROC CONTENTS

- ⌚ Key items to look for:

- Data set name

- # of observations

- # of variables

- Date data set was created and last modified

- List of variables with type, format, and label

# PROC CONTENTS - Example 1

## ○ Syntax:

```
PROC CONTENTS DATA=white;  
RUN;
```

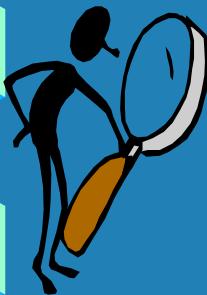
# PROC CONTENTS - Example 1, Output

Data Set Name: WORK.WHITE  
Member Type: DATA  
Engine: V612  
Created: 14:05 Friday, January 26, 2001  
Last Modified: 14:05 Friday, January 26, 2001  
Protection:  
Data Set Type:  
Label:

Observations: 7  
Variables: 8  
Indexes: 0  
Observation Length: 64  
Deleted Observations: 0  
Compressed: NO  
Sorted: NO

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format	Label
5	age	Num	8	16		
6	diab	Num	8	24		Diabetes diagnosis - self-reported
8	diabdiag	Num	8	40		Diabetes diagnosis - lab
4	dob	Num	8	8	DATE9.	Date of birth
7	fgluc	Num	8	32		Fasting glucose
1	gender	Char	8	48		
3	group	Char	8	56		
2	id	Num	8	0		



# What does my Data Look Like?

- ➊ PROC PRINT -> prints a list of observations in a SAS data set
- ➋ Syntax:

PROC PRINT <*options*>;

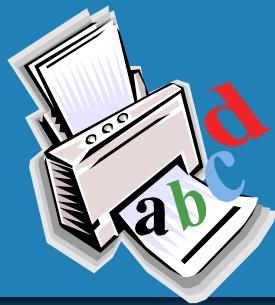
    WHERE *condition*;

    VAR *variable list*;

    BY *variable list*;

    SUM *variable list*;

    RUN;



# PROC PRINT - VAR Statement

- Lists the variables to be printed
- The VAR statement is optional
- If omitted all the variables in the data set will be printed
- Variables are printed in the order listed in VAR statement

# PROC PRINT - Example 2

## ⌚ Syntax:

```
PROC PRINT DATA=white;  
    VAR id gender dob diab;  
RUN;
```

# PROC PRINT - Example 2, Output

Obs	id	gender	dob	diab
1	10	F	01JAN1960	1
2	25	M	02FEB1925	0
3	30	M	03MAR1930	0
4	40	F	04APR1940	1
5	55	U	05MAY1950	1
6	67	F	17FEB1970	0
7	82	F	31AUG1974	0

# PROC PRINT - BY Statement

- Prints data separately for each group in the BY variable
- The BY statement is optional
- When using the BY statement, the data must first be sorted by the variable (s) listed in the BY statement

# PROC PRINT - Example 3

## ○ Syntax:

```
PROC SORT DATA=white;  
    BY diab;  
RUN;
```

```
PROC PRINT DATA=white;  
    VAR id gender age;  
    BY diab;  
RUN;
```

# PROC PRINT - Example 3, Output

diab=0

Obs	id	gender	age
1	25	M	76
2	30	M	70
3	67	F	30
4	82	F	26

diab=1

Obs	id	gender	age
5	10	F	41
6	40	F	60
7	55	U	50

6  
+5  
 

# PROC PRINT - SUM Statement

- Allows variable values to be summed and displayed in output
- The SUM statement is optional
- SUM statement and BY statement can be used together -> variable values will be subtotalized for each BY group
- Summed values will not be saved in SAS data set

# PROC PRINT - Example 4

## ❖ Syntax:

```
PROC PRINT DATA=white;  
  VAR id gender diab;  
  SUM diab;  
RUN;
```

# PROC PRINT - Example 4, Output

Obs	id	gender	diab
1	10	F	1
2	25	M	0
3	30	M	0
4	40	F	1
5	55	U	1
6	67	F	0
7	82	F	0

=====

3



# Key Options to Use in PROC PRINT

- **NOOBS** -> Removes observation numbers from output
- **LABEL** -> Uses variable label as column heading rather than variable name (which is the default)
- **N** -> Prints number of observations at bottom of output
- **OBS =** -> specifies the last observation to be listed
- **FIRSTOBS =** -> specifies the observation number to use as the first observation in listing

# PROC PRINT - Example 5

## ○ Syntax:

```
PROC PRINT DATA=white NOBS N  
      LABEL;  
      VAR id gender diab;  
RUN;
```

# PROC PRINT - Example 5, Output

Diabetes diagnosis		
id	gender	self-reported
10	F	1
40	F	1
67	F	0
82	F	0
25	M	0
30	M	0
55	U	1

N = 7

# PROC PRINT - Example 6

## ⌚ Syntax:

```
PROC PRINT DATA=white LABEL  
  (FIRSTOBS=2 OBS=5);  
  VAR id gender diab;  
RUN;
```

# PROC PRINT - Example 6, Output

Diabetes diagnosis				
Obs	id	gender	self-reported	
2	40	F	1	
3	67	F	0	
4	82	F	0	
5	25	M	0	

# How to Print Only a Subset of the Data

- WHERE statement can be used to display a subset of the data set
- Syntax:

```
PROC PRINT DATA=white NOBS N LABEL;  
    WHERE age < 50;  
    VAR id age gender diab;  
    TITLE "Patients younger than 50";  
RUN;  
TITLE;
```

# PROC PRINT - Example 7, Output

Patients younger than 50

Diabetes diagnosis			
id	age	gender	self-reported
10	41	F	1
67	30	F	0
82	26	F	0

N = 3

# WHERE Statement for Data Cleaning

- WHERE statement can also be very useful when doing data checks

Missing values

Example: WHERE age = .;

Out-of-range values

Example: WHERE age > 100;

Logic checks

Example: WHERE diabdiag=0 and fgluc >= 126;

# How to Obtain Descriptive Statistics

- ➊ Proc Means
- ➋ Proc Summary
- ➌ Proc Univariate
- ➍ Proc Freq

# PROC MEANS

- Provides descriptive statistics for numeric variables (mean, standard deviation, range, min, max, etc.)
- Easy to use
- Other procedures can provide additional descriptive statistics

# PROC MEANS

## ○ Syntax:

```
PROC MEANS <options> <statistic  
keyword list>;  
WHERE condition;  
VAR variable list;  
CLASS variable list;  
BY variable list;  
RUN;
```

# PROC MEANS - Statistic Keywords

- N - # of observations
- NMISS - # of observations with missing values
- MIN - minimum value
- MAX - maximum value
- RANGE - range of values
- SUM - sum of values
- MEAN - mean
- VAR - variance
- STD - standard deviation

Statistics in yellow are printed by default

# PROC MEANS - Example 8

## ○ Syntax:

```
PROC MEANS DATA=white N MEAN  
STD;  
RUN;
```

# PROC MEANS -

## Example 8, Output

Variable	N	Mean	Std Dev
id	7	44.1428571	25.2416889
dob	7	-3618.57	7029.40
age	7	50.4285714	19.2860670
diab	7	0.4285714	0.5345225
fgluc	6	116.8333333	22.4269183
diabdiag	6	0.3333333	0.5163978

# PROC MEANS - Example 9

## ○ Syntax:

```
PROC MEANS DATA=white N MEAN  
    STD;  
    VAR age fgluc;  
RUN;
```

# PROC MEANS - Example 9, Output

Variable	N	Mean	Std Dev
age	7	50.4285714	19.2860670
fgluc	6	116.8333333	22.4269183

# PROC MEANS - CLASS Statement

- CLASS statement -> calculates statistics for each group in CLASS variable
- CLASS variables can be numeric or character
- Data does not need to be sorted when using the CLASS statement

# **PROC MEANS -**

## **Example 10**

- Syntax:

```
PROC MEANS DATA=white N MEAN  
STD;  
CLASS diab;  
VAR fgluc;  
RUN;
```

# PROC MEANS - Example 10, Output

Analysis Variable : fgluc Fasting glucose

Diabetes diagnosis

self-reported	N Obs	N	Mean	Std Dev
0	4	4	103.5000000	11.2101145
1	3	2	143.5000000	2.1213203

**N Obs** -> total number of observations in a subgroup including both the number of missing and number of nonmissing observations

**N** -> number of observations in subgroup with nonmissing values

# PROC SUMMARY

- Computes descriptive statistics on numeric variables and outputs the results to a new data set
- By default PROC SUMMARY does not display any output
- Using the PRINT option will display the output
- Computes the same statistics as PROC MEANS
- Syntax is the same format as PROC MEANS

# PROC UNIVARIATE

- Provides descriptive statistics for numeric variables (mean, standard deviation, range, min, max, etc.)
- Provides more detailed information on the distribution of a variable (extreme values, plots to illustrate distribution, etc)

# PROC UNIVARIATE

## ⌚ Syntax:

```
PROC UNIVARIATE <options>;  
  WHERE condition;  
  VAR variable list;  
  BY variable list;  
RUN;
```



# PROC UNIVARIATE - Key Items

- N - # of observations
- Mean
- Standard deviation
- Variance
- Median
- Upper quartile (75th percentile)
- Lower quartile (25th percentile)
- Mode

# PROC UNIVARIATE -

## Example 11

### ○ Syntax:

```
PROC UNIVARIATE DATA=white;  
  VAR fgluc;  
RUN;
```

# PROC UNIVARIATE - Example 11, Output

Variable=FGLUC		Fasting glucose						
Moments				Quantiles(Def=5)				
N	6	Sum Wgts	6	100% Max	145	99%	145	
Mean	116.8333	Sum	701	75% Q3	142	95%	145	
Std Dev	22.42692	Variance	502.9667	50% Med	110	90%	145	
Skewness	0.464587	Kurtosis	-2.26725	25% Q1	99	10%	95	
USS	84415	CSS	2514.833	0% Min	95	5%	95	
CV	19.19565	Std Mean	9.155751			1%	95	
T:Mean=0	12.76065	Pr> T	0.0001	Range	50			
Num ^= 0	6	Num > 0	6	Q3-Q1	43			
M(Sign)	3	Pr>= M	0.0313	Mode	95			
Sgn Rank	10.5	Pr>= S	0.0313					

# PROC UNIVARIATE - Example 11, Output

Extremes			
Lowest	Obs	Highest	Obs
95	(2)	99	(6)
99	(6)	100	(3)
100	(3)	120	(7)
120	(7)	142	(5)
142	(5)	145	(1)

Missing Value .  
Count 1  
% Count/Nobs 14.29

# PROC UNIVARIATE - Options

- PLOT -> Creates various distribution plots
  - Stem and leaf plot
  - Horizontal bar chart
  - Box plot
  - Side-by-side box plots (if BY statement used)
  - Normal probability plot

# PROC UNIVARIATE -

## Example 12

- Syntax:

```
PROC UNIVARIATE DATA=white  
    PLOT;  
    VAR age;  
RUN;
```

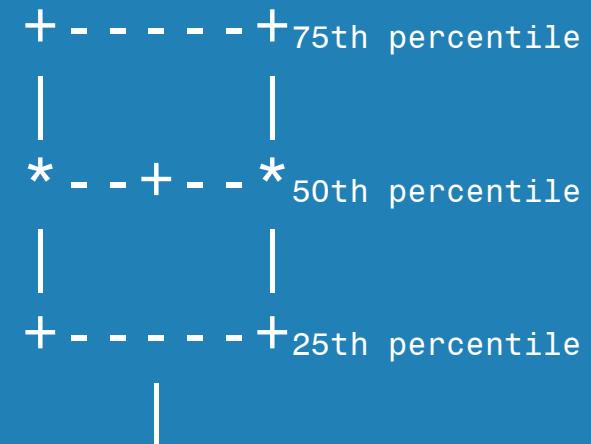
# PROC UNIVARIATE - Example 12, Output

## • AGE

Stem Leaf	#
7 06	2
6 0	1
5 0	1
4 1	1
3 0	1
2 6	1

Multiply Stem.Leaf by 10\*\*+1

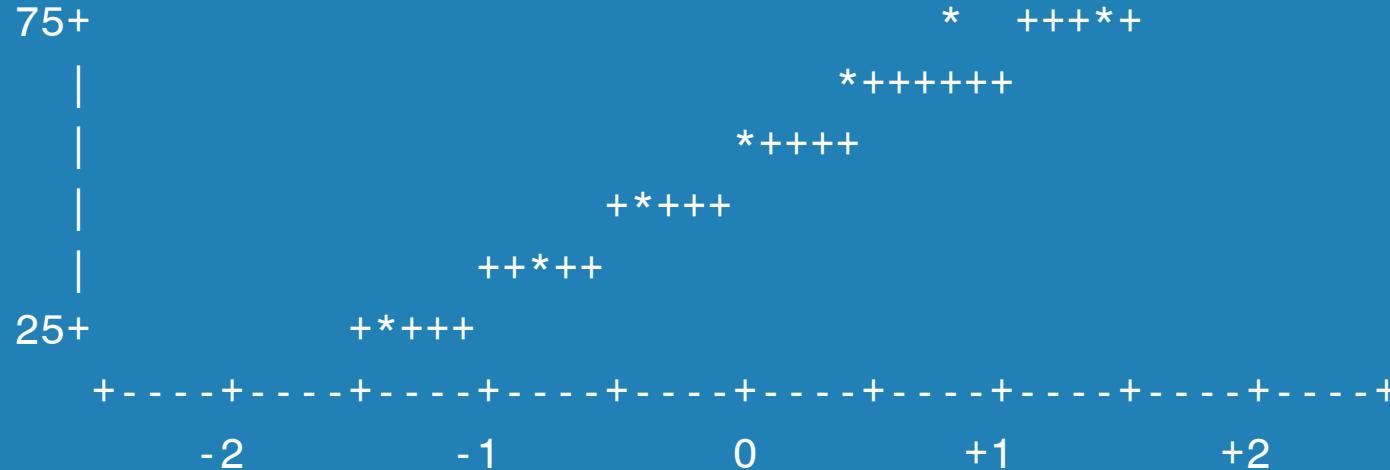
## Boxplot



+ = sample mean

# PROC UNIVARIATE - Example 12, Output

## Normal Probability Plot

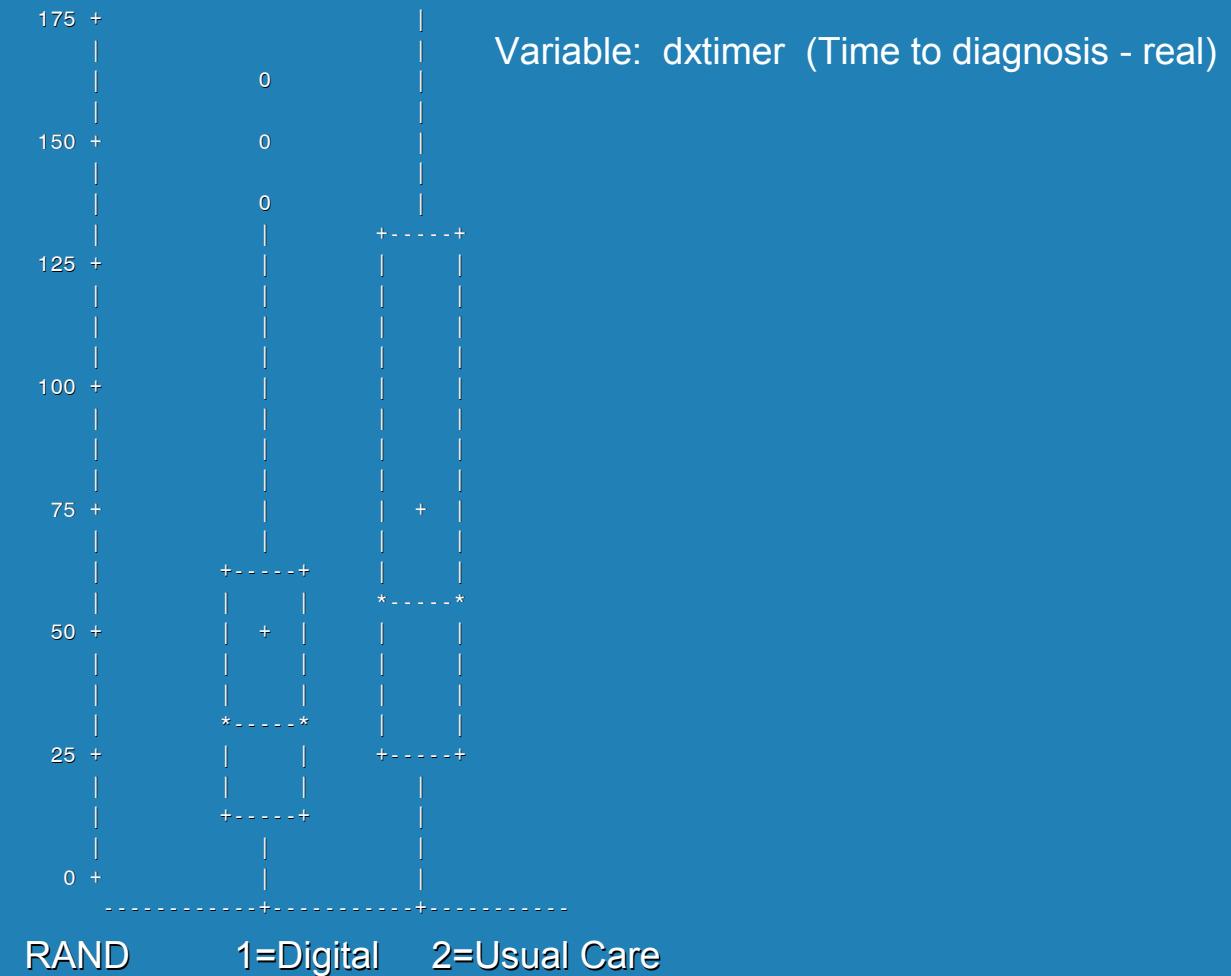


\* - data values

+ - reference straight line

If data are normal, asterisks should lie on reference line

# PROC UNIVARIATE - Example 13, Output



# Descriptive Statistics - Categorical Variables

- PROC FREQ
  - 1) Provides descriptive statistics in the form of frequencies and crosstabulation tables
  - 2) Provides statistics to analyze the relationships between variables
- We will only be covering number 1 in this presentation

# PROC FREQ

- Provides various forms of crosstabulation tables

One-way frequencies -> generates a table with the frequency of the different values of a variable  
Two-way crosstabulation table -> generates a frequency table with the values of the two variables  
N-way crosstabulation table -> generates a n-way frequency table with the values of the n variables

# PROC FREQ

## ❖ Syntax:

PROC FREQ <*options*>;

WHERE *condition*;

BY *variable list*;

TABLES *variable list* </*options*>;

RUN;

- ❖ If TABLES statement is omitted, one-way tables will be generated for all variables

# PROC FREQ - TABLES Statement

- ⦿ One-way frequency table -> list the variables separated by a space
- ⦿ Syntax:

```
PROC FREQ DATA=white;  
    TABLES gender diab;  
RUN;
```

# PROC FREQ - Example 14, Output

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	4	57.1	4	57.1
M	2	28.6	6	85.7
U	1	14.3	7	100.0

DIAB	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	4	57.1	4	57.1
1	3	42.9	7	100.0

# PROC FREQ - TABLES Statement

- ➊ Two-way crosstab table -> var1\*var2
  - First variable - generates the rows of table
  - Second variable - generates the columns of table

- ➋ Syntax:

```
PROC FREQ DATA=white;
```

```
WHERE gender ne 'U';
```

```
TABLES gender*diab;
```

```
RUN;
```

# PROC FREQ - Example 15, Output

GENDER		DIAB(Diabetes diagnosis self-reported)			
Frequency					
Percent					
Row Pct					
Col Pct		0	1	Total	
F		2	2	4	
33.33		33.33		66.67	
50.00		50.00			
50.00		100.00			
M		2	0	2	
33.33		0.00		33.33	
100.00		0.00			
50.00		0.00			
Total		4	2	6	
66.67		33.33		100.00	

# PROC FREQ - TABLES

## Statement Options

- **LIST** -> displays output in a list format rather than in a table format
- **MISSING** -> missing values are interpreted as a nonmissing response and included in calculations of percentages
- **NOCOL** -> suppresses column percentages in table
- **NOROW** -> suppresses row percentages in table

# PROC FREQ - TABLES

## Statement Options

- **NOCUM** -> suppresses cumulative frequencies and percentages for one-way frequencies
- **NOFREQ** -> suppresses cell counts for a table and counts for row totals
- **NOPERCENT** -> suppresses cell percentages and percentages for row and column totals in table

# PROC FREQ - Example 16

## ⌚ Syntax:

```
PROC FREQ DATA=white;  
  TABLES gender*diabdiag/MISSING NOCOL  
        NOROW;  
  RUN;
```

# PROC FREQ - Example 16, Output

GENDER		DIABDIAG(Diabetes diagnosis-lab)			Total
	Frequency	.	0	1	
F	Percent				
	14.29	1	2	1	4 57.14
M	Percent				
	0.00	0	2	0	2 28.57
U	Percent				
	0.00	0	0	1	1 14.29
Total		14.29	57.14	28.57	100.00

# PROC FREQ - Example 17

- LIST and MISSING options can be useful when creating new variables
- Can be used to ensure that the new variable is coded correctly
- Syntax:

PROC FREQ DATA=white;

TABLES fgluc\*diabdiag/LIST MISSING;

RUN;

# PROC FREQ - Example 17, Output

FGLUC	DIABDIAG	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	.	1	14.3	1	14.3
95	0	1	14.3	2	28.6
99	0	1	14.3	3	42.9
100	0	1	14.3	4	57.1
120	0	1	14.3	5	71.4
142	1	1	14.3	6	85.7
145	1	1	14.3	7	100.0

# PROC FREQ - Options

- ORDER -> indicates the order the variable values are shown in table

DATA - order of values as encountered in input data set

FORMATTED - order as specified by formatted values

FREQ - order of values with most observations

INTERNAL - order as specified by unformatted values (default)

# **PROC FREQ -**

## **Example 18**

- Syntax:

```
PROC FREQ DATA=white  
    ORDER=FREQ;  
    TABLES gender;  
    TITLE "Gender ordered by freq";  
RUN;  
TITLE;
```

# PROC FREQ - Example 18, Output

Gender ordered by freq

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	4	57.1	4	57.1
M	2	28.6	6	85.7
U	1	14.3	7	100.0

# PROC FREQ - TABLES Statement

- N-way crosstab table ->var1\*var2\*...\*varN
  - Last variable - generates the columns of table
  - Next to last variable - generates the rows of table
  - Combination of remaining variables - generates stratum
- Syntax:

```
PROC FREQ DATA=white;  
  TABLES var1*var2*var3*...*varN;  
  RUN;
```



# How to Plot Data

- PROC PLOT -> provides simple plots of two variables
- Syntax:

```
PROC PLOT <options>;  
  WHERE condition;  
  BY variable list;  
  PLOT variable list </options>;  
RUN;
```

# PROC PLOT

- PLOT var1\*var2;

Var1 will be on the vertical axis

Var2 will be on the horizontal axis

By default, A,B, and C are used as plotting symbols

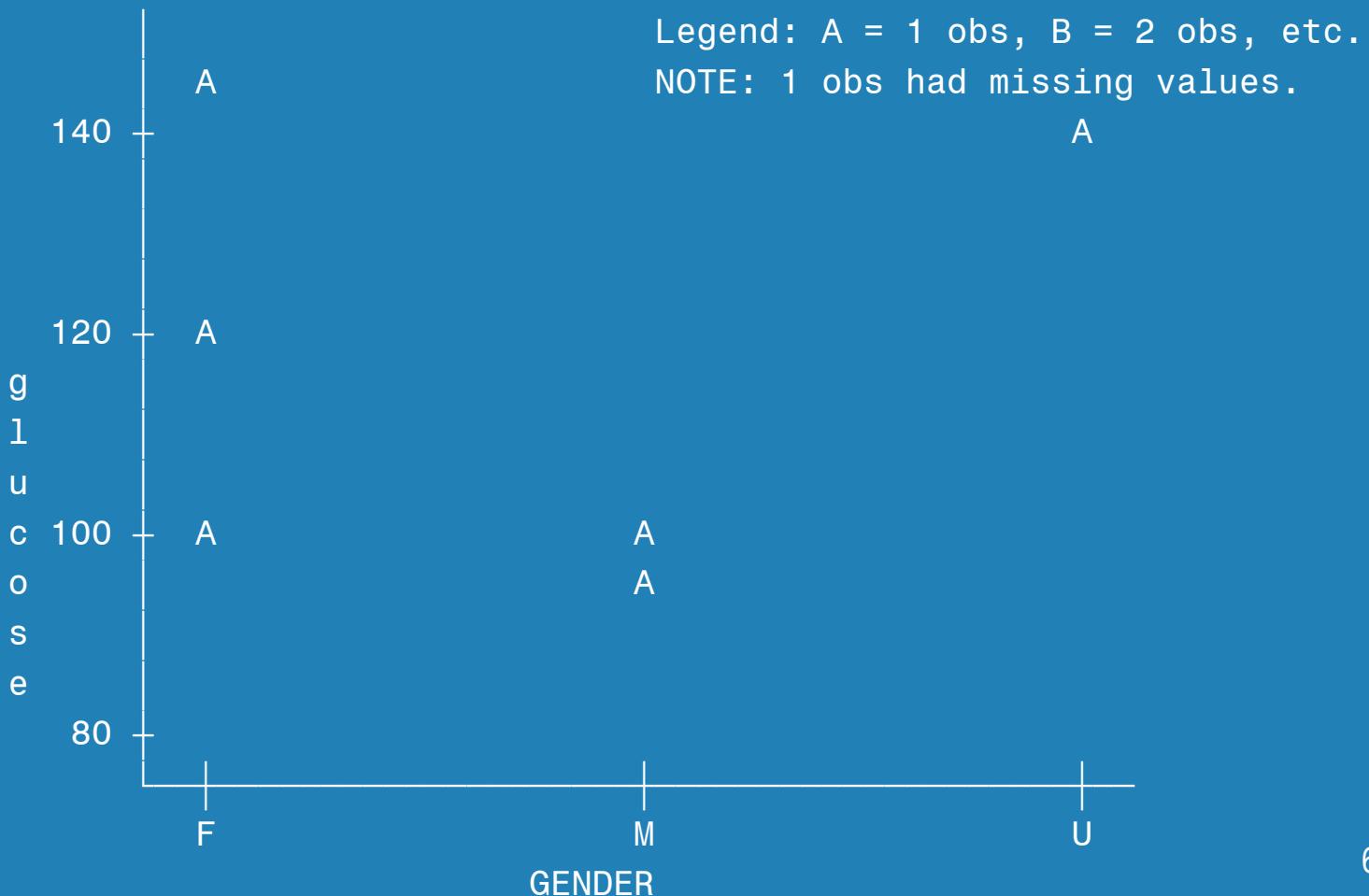
- Syntax:

PROC PLOT DATA=white;

PLOT fgluc\*gender;

RUN;

# PROC PLOT - Example 19, Output



# PROC PLOT

- Plotting symbols can be customized
- **PLOT var1\*var2='\*';**  
Specifies the plotting symbol to be an asterisk
- **PLOT var1\*var2=var3;**  
Specifies the plotting symbol to be the values of var3  
Var3 can be numeric or character

# PROC PLOT - Options

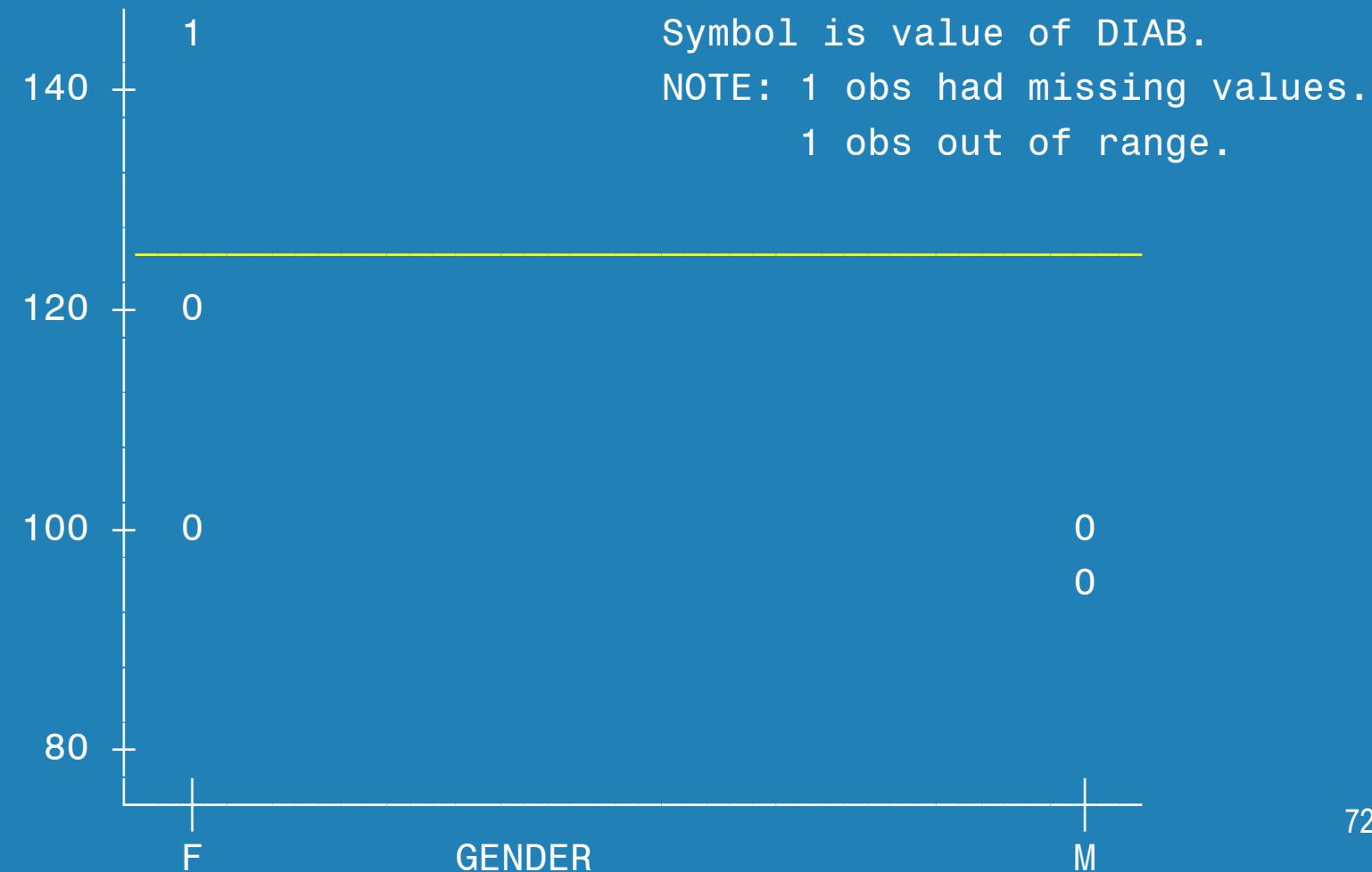
- ⦿ **HAXIS (VAXIS)** -> indicates values to use as tick marks of the horizontal (vertical) axis
- ⦿ **HZERO (VZERO)** -> specifies the value of 0 for the first tick mark on axis
- ⦿ **HREF (VREF)** -> draws a reference line on the plot perpendicular to the horizontal (vertical) axis
- ⦿ **OVERLAY** -> overlays all plots of a PLOT statement on the same set of axes (PLOT a\*b c\*d/overlay;)

# PROC PLOT - Example 20

## ○ Syntax:

```
PROC PLOT DATA=white;  
  PLOT fgluc*age=diab/HAXIS='F' 'M' VREF=126;  
RUN;
```

# PROC PLOT - Example 20, Output





## For More Information

- SAS Procedures Guide - Version 6
- SAS Help System in Version 6.12
- SAS Tech support -  
[www.sas.com/service/techsup/intro.html](http://www.sas.com/service/techsup/intro.html)
- SAS System for Elementary Statistical Analysis by Schlotzhauer and Littell